

KR04/0821

PCT/KR 2004/000821

RO/KR 13. 04. 2004



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Intellectual  
Property Office.

출원 번호 : 10-2003-0022784  
Application Number

출원 년 월 일 : 2003년 04월 10일  
Date of Application APR 10, 2003

출원인 : 에스케이 텔레콤주식회사  
Applicant(s) SK TELECOM CO., LTD.

**PRIORITY  
DOCUMENT**

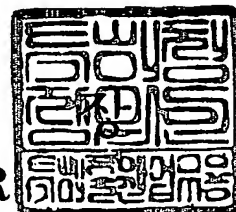
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)



2004 년 04 월 13 일

특 허 청

COMMISSIONER



## 【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【제출일자】	2003.04.10
【발명의 명칭】	비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법 및 장치
【발명의 영문명칭】	Method and Apparatus for Determining Search Range for Adaptive Motion Vector for Use in Video Encoder
【출원인】	
【명칭】	에스케이텔레콤 주식회사
【출원인코드】	1-1998-004296-6
【대리인】	
【성명】	이철희
【대리인코드】	9-1998-000480-5
【포괄위임등록번호】	2000-010209-0
【대리인】	
【성명】	송해모
【대리인코드】	9-2002-000179-4
【포괄위임등록번호】	2002-031289-6
【발명자】	
【성명의 국문표기】	황인성
【성명의 영문표기】	HWANG, In Seong
【주민등록번호】	690410-1231629
【우편번호】	137-070
【주소】	서울특별시 서초구 서초동 1346 현대아파트 10동 807호
【국적】	KR
【발명자】	
【성명의 국문표기】	김철우
【성명의 영문표기】	KIM, Chul Woo
【주민등록번호】	680819-1840410
【우편번호】	138-240
【주소】	서울특별시 송파구 신천동 7 (5/3) 장미아파트 5-908
【국적】	KR

## 【발명자】

【성명의 국문표기】

홍민철

【성명의 영문표기】

HONG, Min Cheol

【주민등록번호】

640530-1141010

【우편번호】

121-773

【주소】

서울특별시 마포구 도화2동 현대2차아파트 210동 1301호

【국적】

KR

## 【발명자】

【성명의 국문표기】

인경석

【성명의 영문표기】

IN, Kyoung Seok

【주민등록번호】

721104-1319614

【우편번호】

463-010

【주소】

경기도 성남시 분당구 정자동 81-2 101호

【국적】

KR

## 【심사청구】

청구

## 【취지】

특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인  
이철희 (인) 대리인  
송해모 (인)

## 【수수료】

【기본출원료】

20 면 29,000 원

【가산출원료】

8 면 8,000 원

【우선권주장료】

0 건 0 원

【심사청구료】

22 항 813,000 원

【합계】

850,000 원

## 【첨부서류】

1. 요약서·명세서(도면)\_1통

**【요약서】****【요약】**

본 발명은 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법 및 장치에 관한 것이다.

본 발명은 화상의 연속성을 나타내는 입력 화상 신호가 공급되며, 입력 화상 신호의 각각의 화상을 매크로 블록들로 분할하여 매크로 블록의 움직임 벡터의 움직임 추정을 통해 영상을 부호화하는 비디오 부호기에서, 우선 현재 매크로 블록에 인접한 매크로 블록의 수를 판단한다. 판단 결과, 인접한 매크로 블록의 수가 2개 이상인 경우 인접한 매크로 블록의 움직임 벡터의 크기를 구하여 그 중 움직임이 가장 큰 매크로 블록의 움직임 벡터를 결정한다. 현재 매크로 블록의 적응 움직임 벡터가 가질 수 있는 최소 탐색 영역값을 정의하여 그 값과 움직임이 가장 큰 인접 매크로 블록의 움직임 벡터의 크기를 비교하여 더 큰 값을 적응 움직임 벡터의 탐색 영역값으로 결정한다. 다음으로, 적응 움직임 벡터의 탐색 영역값과 사용자가 정의한 적응 움직임 벡터의 탐색 영역값을 비교하여 더 작은 값을 최종적인 적응 움직임 벡터의 탐색 영역값으로 결정하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법 및 장치를 제공한다.

**【대표도】**

도 6

**【색인어】**

비디오 부호기, 적응 움직임 벡터, FBMA, MPEG, H.26X, JVT

## 【명세서】

## 【발명의 명칭】

비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법 및 장치{Method and Apparatus for Determining Search Range for Adaptive Motion Vector for Use in Video Encoder}

## 【도면의 간단한 설명】

도 1은 종래 여러 동영상 부호화 표준인 H.261, MPEG-1 또는 MPEG-2 등에서 이용되고 있는 움직임 벡터의 추정 방법에서 현재 부호화할 움직임 벡터와 그 주변의 움직임 벡터를 나타내는 도면,

도 2는 종래 여러 동영상 부호화 표준인 H.263 또는 MPEG-4 등에서 이용되고 있는 움직임 벡터의 추정 방법에서 현재 부호화할 움직임 벡터와 그 주변의 움직임 벡터를 나타내는 도면,

도 3은 종래 여러 동영상 부호화 표준인 H.263 또는 MPEG-4에서 중간값 예측 기법을 이용하여 현재 매크로 블록의 움직임 벡터의 탐색 영역을 구하는 방법을 설명하기 위한 도면,

도 4는 종래 여러 동영상 부호화 표준인 H.263 또는 MPEG-4에서 1차원 최소 비트량 예측 기법을 이용하여 현재 매크로 블록의 움직임 벡터를 부호화할 때의 비트 스트림(Bit-stream) 구조,

도 5는 본 발명의 바람직한 실시예에 따른 현재 부호화하고자 하는 매크로 블록과 그 인접한 매크로 블록의 영역을 나타내는 도면,

도 6은 본 발명의 바람직한 실시예에 따른 현재 부호화하고자 하는 매크로 블록의 움직임 벡터의 탐색 영역을 구하는 과정을 나타내는 순서도이다.

#### 【발명의 상세한 설명】

#### 【발명의 목적】

#### 【발명이 속하는 기술분야 및 그 분야의 종래기술】

- <➤ 본 발명은 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법 및 장치에 관한 것이다. 더욱 상세하게는, 화상의 연속성을 나타내는 입력 화상 신호가 공급되며 입력 화상 신호의 화상을 다수의 매크로 블록(Macro Block : MB)으로 분할하여 매크로 블록의 움직임 벡터의 움직임 탐색을 통해 영상을 부호화하는 비디오 부호기에서, 움직임 벡터가 가질 수 있는 영역을 인접한 매크로 블록의 움직임 벡터로부터 추정하여 적응 움직임 벡터의 탐색 영역을 결정하는 방법 및 장치에 관한 것이다.
- <➤ 근래에, 동영상 혹은 정지 영상을 압축 부호화하기 위한 다양한 기술들과 표준안들이 제시되고 있으며, 이러한 표준안들에서는 보다 효율적이고 높은 성능으로 영상을 압축 부호화하는 것을 궁극적인 목적으로 하고 있다. 현재 동영상 혹은 정지 영상 등의 영상 부호화에 관련된 표준안으로 MPEG-1, MPEG-2, MPEG-4, H.261, H.263 및 JVT 방식 등이 제안되었거나 제안되고 있다.
- <➤ 이와 같은 여러 국제 표준들과 같이 JVT 방식에서도 움직임 추정을 행하고 있고, 이 결과로 나타난 잔류 신호(Residual Signal)를 부호화하기 위해서 영상 신호를 시공간적으로 변환(Transform)하고 지그재그 스캐닝(Zig-Zag Scanning)에 이

은 양자화(Quantization)의 과정을 이용하여 부호화를 수행한다. 이와 같은 영상 부호화에 있어서는 블록 기반으로 영상을 부호화한다. 이처럼, 블록을 기반으로 한 움직임 추정을 통하여 동영상 혹은 정지 영상을 압축 부호화함에 있어서, 영상 프레임(Frame) 내의 블록들이 합쳐져 구성되는 매크로 블록은 하나의 움직임 벡터를 가지게 된다. 다수의 매크로 블록 중 인접한 매크로 블록의 움직임 벡터 사이에는 어느 정도의 상관성이 존재하게 되므로 각 매크로 블록의 움직임 벡터를 부호화하기 위해서 인접한 매크로 블록들의 이전 프레임의 움직임 벡터를 이용한다. 이 방법은, 현재 부호화할 매크로 블록에 대한 움직임 벡터 예측치(Prediction Vector)를 구하고 이것과 인접한 매크로 블록들의 움직임 벡터와의 차(Difference Vector)를 부호화하는 방법으로서, 블록 기반 영상 부호화 방법에서 효율적인 부호화 기법으로 이용되고 있다.

<10> 그러나, 종래의 움직임 벡터의 추정 알고리즘은 그 개념은 간단하지만, 움직임 벡터를 찾기 위해 요구되는 계산 과정이 매우 복잡하다. 따라서 움직임 벡터를 찾는 과정은 전체 부호화 과정에 소요되는 시간의 대부분을 차지했다. 가장 일반적인 움직임 추정 알고리즘인 전 영역 탐색(Full Block Matching Algorithm : FBMA) 방법은 탐색 영역 안에 있는 모든 매크로 블록들에 대해서 왜곡 치수(Distortion Measure)를 구한 다음, 최소의 왜곡을 나타내는 매크로 블록을 찾는 방법이다. FBMA 방법은 움직임 벡터로 결정될 가능성이 있는 움직임 벡터를 포함하는 모든 매크로 블록과 현재 움직임 벡터를 구하고자 하는 매크로 블록에 대해서 탐색을 행하기 때문에 움직임 벡터를 찾는 데 있어서 매우 최적화된 방법이기도 하다. 그렇지만 과도한 왜곡 계산은 비디오 부호화기에 상당한 부담을 주기 때문에 실제적인 응용이 제한적일 수밖에 없다. 예컨대, 고선명 텔레비전과 같이 탐색 영역이 넓은 경우, 비교해야 할 화소수가 엄청나게 많으므로 실시간 처리가 거의 불가능하며, 실시간 처리가 가능하게 하기 위해서는 하드웨어의 크기와 용량이 엄청나게 증가하는 문제가 발생한다.

- <11> 도 1은 종래 여러 동영상 부호화 표준인 H.261, MPEG-1 또는 MPEG-2 등에서 이용되고 있는 움직임 벡터의 추정 방법에서 현재 부호화할 움직임 벡터와 그 주변의 움직임 벡터를 나타내는 도면이다.
- <12> H.261, MPEG-1 또는 MPEG-2 등에서 이용되고 있는 움직임 벡터의 추정 방법은, 현재 매크로 블록의 움직임 벡터 MV(Motion Vector)를 추정할 때, 직전 매크로 블록의 움직임 벡터 MV1 만을 예측치(Predictor)로 사용하는 방법이다. 이 방법은 직전 매크로 블록의 움직임 벡터 만을 예측치로 사용하기 때문에 현재 매크로 블록의 움직임 벡터를 계산하는데 있어서 구해진 값의 신뢰도를 보장하기 어렵다는 문제점이 있다.
- <13> 도 2는 종래 여러 동영상 부호화 표준인 H.263 또는 MPEG-4 등에서 이용되고 있는 움직임 벡터의 추정 방법에서 현재 부호화할 움직임 벡터와 그 주변의 움직임 벡터를 나타내는 도면이다.
- <14> H.263 또는 MPEG-4 등에서 이용되고 있는 움직임 벡터의 추정 방법은, 도 2에 도시된 바와 같이, 현재 부호화할 움직임 벡터 MV의 주변에 좌측과 상측 및 우상측으로 인접한 3개의 매크로 블록의 각각 3개의 움직임 벡터 MV1, MV2 및 MV3를 예측에 사용할 후보로 선택한다. H.263 또는 MPEG-4 등에서 이용되고 있는 움직임 벡터의 추정 방법에는 중간값 예측 기법 및 최소 비트량 예측 기법이 있다.
- <15> 도 3은 종래 여러 동영상 표준인 H.263이나 MPEG-4에서 중간값 예측 기법을 이용하여 현재 매크로 블록의 움직임 벡터의 탐색 영역을 구하는 방법을 설명하기 위한 도면이다.
- <16> 예측기(PREDICTOR)(301)는 도 2에서 보는 바와 같은 현재 부호화할 움직임 벡터 MV의 주변에 있는 3개의 움직임 벡터 MV1, MV2 및 MV3에 대해서 x, y 각 방향의 값들 중 중간값을 선



택하여  $x$ ,  $y$  각 요소별로 출력한다.  $x$ ,  $y$  각 요소별로 출력한 중간값들은 수학적 식 1과 같이 표현할 수 있다.

<17> 【수학적 식 1】  $PMVx = \text{Median}(MV1x, MV2x, MV3x)$ ,  $PMVy = \text{Median}(MV1y, MV2y, MV3y)$

<18> 그리고 연산기(302)는 상기  $PMVx$ ,  $PMVy$  값과 현재 부호화할 움직임 벡터의 차  $MVD$ 를 구하여  $x$ ,  $y$  각 요소별로 디코더(Decoder)에 보내준다.  $x$ ,  $y$  각 요소별로 표현되는 차이값  $MVDx$  및  $MVDy$ 는 수학적 식 2와 같이 표현할 수 있다.

<19> 【수학적 식 2】  $MVDx = MVx - PMVx$ ,  $MVDy = MVy - PMVy$

<20> 여기서  $MVx$ 는 현재 부호화할 움직임 벡터의  $x$  요소이고,  $MVDy$ 는 현재 부호화할 움직임 벡터의  $y$  요소이다.

<21> 중간값 예측 기법은 인접한 세 매크로 블록의 움직임 벡터를 이용하기 때문에 상대적으로 도 1의 방법보다는 신뢰도가 높다. 중간값 예측 기법은 3개의 인접한 매크로 블록의 움직임 벡터를 동등하게 이용한다는 특징이 있는데, 인접한 매크로 블록들의 움직임 벡터들은 현재 블록의 움직임 벡터와의 상관성의 정도에 있어서 실제로 차이를 갖는다. 따라서 이러한 차이를 고려하지 않았다는 측면에서 볼 때, 중간값 예측 기법도 절대적으로 신뢰도가 낮다는 문제점이 있다.

<22> 이러한 문제점을 극복하기 하여 고안된 것이 최소 비트량 예측 기법을 이용하여 현재 매크로 블록의 움직임 벡터의 탐색 영역을 구하는 방법이다. 이 방법에서는 현재 부호화할 움직임 벡터와 주변의 움직임 벡터들의 위치에 대하여 예측 오차를 구하고, 그 예측 오차의 비트 수가 가장 적은 주변 움직임 벡터를 선택한다. 예측 오차의 비트 수가 가장 적은 주변 움직임

벡터를 선택한 후, 현재 부호화할 움직임 벡터와 선택한 주변 움직임 벡터와의 예측 오차를 부호화하고, 어떤 움직임 벡터가 예측에 사용되었는지를 알려주는 모드(MODE) 정보를 보내준다.

<23> 도 4는 종래 여러 동영상 부호화 표준인 H.263 또는 MPEG-4에서 1차원 최소 비트량 예측 기법을 이용하여 현재 매크로 블록의 움직임 벡터를 부호화할 때의 비트 스트림(Bit-stream) 구조이다.

<24> 도 4에 나타낸 바와 같이 1차원 최소 비트량 예측 기반의 움직임 벡터 부호화는 2차원 상에서 x 요소(Component)에 대한 예측 오차  $MVD_{x\_mbp}$ , x 모드 정보  $MODE_x$ , y 성분에 대한 예측 오차  $MVD_{y\_mbp}$  및 y 모드 정보  $MODE_y$ 로 각각 이루어지고 있다.

<25> 이러한 최소 비트량 예측에 기반을 둔 부호화의 경우에는, 전송되는 데이터에 예측 오차 정보인 MVD 정보뿐만 아니라 모드 정보까지 x, y 각 요소에 대해서 존재하고 있다. 예측 오차 정보는 최소 비트량 예측 기법이 적용되므로 적은 비트 수를 차지하지만, 이에 비해서 모드 정보는 많은 비트수를 차지하기 때문에 부담은 상대적으로 커지게 된다. 이러한 부담은 실시간 압축 부호화 처리에 장애가 될 수 있고, 전송되는 데이터의 양도 증가할 우려가 높아 효율적이지 못하다는 문제점이 있다.

#### 【발명이 이루고자 하는 기술적 과제】

26> 상기한 문제점을 해결하기 위해 본 발명은, 현재 매크로 블록의 움직임 벡터가 이미 부호화된 인접한 매크로 블록의 움직임 벡터의 움직임 경향과 유사성이 크다는 특성을 이용하여, 움직임 벡터의 탐색 영역을 인접한 매크로 블록의 정보로부터 적응적으로 추정한다. 이는 인접한 매크로 블록의 최대 움직임 벡터의 크기를 구하고, 현재 매크로 블록의 움직임 벡터가 가질

수 있는 최소 영역값을 정의한다. 그 후, 인접한 매크로 블록의 최대 움직임 벡터의 크기와 현재 매크로 블록의 움직임 벡터가 가질 수 있는 최소 영역값을 비교하여 변환하는 과정을 통해 현재 매크로 블록이 가질 수 있는 움직임 벡터의 탐색 영역을 정의함으로써, 부호화부의 계산량을 절감할 수 있는 방법 및 시스템을 제공하는 것을 목적으로 한다.

### 【발명의 구성】

<27>      상기한 목적을 달성하기 위해 본 발명은 화상의 연속성을 나타내는 입력 화상 신호가 공급되며 상기 입력 화상 신호의 각각의 화상을 다수의 매크로 블록으로 분할하여 상기 매크로 블록의 움직임 벡터의 움직임 추정을 통해 영상을 부호화하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법에 있어서, (a) 현재 매크로 블록에 인접한 인접 블록의 수를 판단하는 단계; (b) 상기 단계 (a)에서의 판단 결과, 상기 인접 블록의 수가 2개 이상인 경우, 상기 인접 블록의 움직임 벡터의 크기를 구하여 그 중 움직임이 가장 큰 블록의 움직임 벡터를 결정하는 단계; (c) 상기 현재 매크로 블록의 상기 적응 움직임 벡터가 가질 수 있는 최소 탐색 영역값을 정의하는 단계; (d) 상기 단계 (b)에서 결정된 상기 움직임이 가장 큰 블록의 움직임 벡터 크기의 두 배와 상기 단계 (c)에서 구한 상기 적응 움직임 벡터의 최소 탐색 영역값을 비교하여 더 큰 값을 상기 적응 움직임 벡터의 탐색 영역값으로 결정하는 단계; 및 (e) 상기 단계 (d)에서 구한 상기 적응 움직임 벡터의 탐색 영역값과 사용자가 정의한 적응 움직임 벡터의 탐색 영역값을 비교하여 더 작은 값을 최종 적응 움직임 벡터의 탐색 영역값으로 결정하는 단계를 포함하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법을 제공하는 것을 특징으로 한다.

<28>      또한, 본 발명의 다른 목적에 의하면, 화상의 연속성을 나타내는 입력 화상 신호가 공급되며 상기 입력 화상 신호의 각각의 화상을 다수의 매크로 블록으로 분할하여 상기 매크로 블

특의 움직임 벡터의 움직임 추정을 통해 영상을 부호화하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 영상 압축 장치에 있어서, 현재 매크로 블록에 인접한 블록의 수를 판단하는 판단부; 상기 매크로 블록의 움직임 벡터의 크기 또는 상기 현재 매크로 블록의 상기 적응 움직임 벡터가 가질 수 있는 최소 탐색 영역값을 계산하는 계산부; 상기 계산부에서 계산된 인접한 블록들의 움직임 벡터 중 그 크기가 가장 큰 움직임 벡터의 크기의 두 배와 상기 계산부에서 계산한 상기 적응 움직임 벡터의 최소 탐색 영역값을 비교하는 제 1 비교부; 상기 제 1 비교부에서의 비교 결과, 더 큰 값을 상기 적응 움직임 벡터의 탐색 영역값으로 결정하는 결정부; 상기 계산부에서 계산한 상기 적응 움직임 벡터의 탐색 영역값과 사용자가 정의한 적응 움직임 벡터의 탐색 영역값을 비교하는 제 2 비교부; 및 상기 제 2 비교부에서의 비교 결과, 더 작은 값을 최종 적응 움직임 벡터의 탐색 영역값으로 결정하는 최종 결정부를 포함하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 영상 압축 장치를 제공하는 것을 특징으로 한다.

29> 이하, 본 발명의 바람직한 실시예를 첨부된 도면들을 참조하여 상세히 설명한다. 우선 각 도면의 구성요소들에 참조부호를 부가함에 있어서, 동일한 구성요소들에 대해서는 비록 다른 도면상에 표시되더라도 가능한 한 동일한 부호를 가지도록 하고 있음에 유의해야 한다. 또한, 본 발명을 설명함에 있어, 관련된 공지 구성 또는 기능에 대한 구체적인 설명이 본 발명의 요지를 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명은 생략한다.

30> JVT 동영상 압축 방식에서는 부호화 과정에서 시간적, 공간적인 과잉 정보를 제거하는 기법을 사용하여 시간적, 공간적으로 압축된 정보를 생성하고, 이 외에 복호화에 필요한 정보를 부가하여 전송한다. 복호화 과정은 부호화 과정의 역순으로 구성되어 있다. 본 발명에서는 움직임 벡터의 탐색 영역을 인접한 매크로 블록의 정보로부터 적응적으로 추정하여 부호화부의

계산량을 절감할 수 있는 방법 및 장치를 제공한다. 특히, 현재 매크로 블록의 움직임 벡터는 이미 부호화된 인접한 매크로 블록의 움직임 벡터의 움직임 경향과 유사성이 크다는 특성을 이용하여 현재 매크로 블록의 움직임 벡터의 최대 탐색 영역을 효과적으로 결정할 수 있도록 한다.

- <31> 도 5는 본 발명의 바람직한 실시예에 따른 현재 부호화하고자 하는 매크로 블록과 그 인접한 매크로 블록의 영역을 나타내는 도면이다.
- <32> 현재 부호화하고자 하는 매크로 블록을 E라 하고 그 인접한 영역의 블록을 A, B, 및 C라 정의한다. 그리고 블록 A, B 및 C의 움직임 벡터를 각각  $(mv\_a\_x, mv\_a\_y)$ ,  $(mv\_b\_x, mv\_b\_y)$ , 및  $(mv\_c\_x, mv\_c\_y)$ 라 가정한다. 여기서 첨자 x 및 y는 수평 및 수직 방향으로의 움직임 벡터를 각각 의미한다. 이러한 가정 하에, 인접한 영역의 블록 A, B 및 C의 움직임 벡터로부터 현재 부호화하고자 하는 매크로 블록 E가 취할 수 있는 최대 움직임 벡터의 탐색 영역을 구한다.
- <33> 도 6은 본 발명의 바람직한 실시예에 따른 현재 부호화하고자 하는 매크로 블록의 움직임 벡터의 탐색 영역을 구하는 과정을 나타내는 순서도이다.
- <34> 우선, 도 5에서 사용 가능한 인접 블록의 수를 판단한다(S600).
- <35> 도 1에서 설명한 것과 같이, 사용 가능한 인접 블록의 수가 1개 이하인 경우에는 바로 이전 블록의 움직임 벡터만을 예측치로 사용하는 것이기 때문에 현재 매크로 블록의 움직임 벡터를 계산하는 데 있어서 구해진 값의 신뢰도를 보장하기 어렵다. 따라서 사용 가능한 인접 블록의 수가 1개 이하인 경우에는 최종 적응 움직임 벡터의 탐색 영역을 인코딩 시작시 사용자가

디폴트(Default) 값으로 정해놓은 값으로 한다(S616). 단계 S616은 수학적 식 3과 같이 표현할 수 있다.

<36> 【수학적 식 3】  $\text{new\_search\_range\_i} = \text{input\_search\_range}$

<37> 여기서  $\text{new\_search\_range\_i}$  는 현재 부호화하고자 하는 매크로 블록의 최종 적응 움직임 벡터의 탐색 영역을 나타내고,  $\text{input\_search\_range}$  는 오류 방지를 위해, 인코딩 시작시 사용자가 디폴트 값으로 정해놓은 값을 의미한다.

<38> 한편, 사용 가능한 인접 블록의 수가 2개 이상인 경우에는 현재 부호화하고자 하는 매크로 블록 E의 적응 움직임 벡터의 최대 탐색 영역을 결정하기 위하여, 사용 가능한 인접 블록의 움직임 벡터의 크기를 구한다(S602). 각각의 크기 중 가장 큰 값 즉, 사용 가능한 인접 블록의 움직임 벡터 중 움직임이 가장 큰 블록의 움직임 벡터의 크기를 구하여 그 값을 현재 부호화하고자 하는 매크로 블록 E의 수평 및 수직 방향에 대한 적응 움직임 벡터의 최대 탐색 영역으로 결정한다(S604). 단계 S604의 동작은 수학적 식 4와 같이 표현할 수 있다.

<39> 【수학적 식 4】  $\text{max\_MV\_E\_i} = \max[\text{abs}(\text{MV\_A\_i}), \max\{\text{abs}(\text{MV\_B\_i}), \text{abs}(\text{MV\_C\_i})\}]$

<40> 여기서  $\text{max\_MV\_E\_i}$ 는 현재 부호화하고자 하는 매크로 블록 E의 수평 및 수직 방향에 대한 적응 움직임 벡터의 최대 탐색 영역을 나타내는 벡터로  $(\text{max\_MV\_E\_x}, \text{max\_MV\_E\_y})$ 를 나타낸다.  $\max(u, v)$ 는  $u$  및  $v$ 의 최대값을 나타내는 함수이며,  $\text{abs}()$ 는 절대값 함수를 나타낸다. 또한,  $\text{MV\_A\_i}$ 는 인접한 A 영역의 블록의 움직임 벡터인  $(\text{mv\_a\_x}, \text{mv\_a\_y})$ ,  $\text{MV\_B\_i}$ 는 인접한 B 영역의 블록의 움직임 벡터인  $(\text{mv\_b\_x},$

mv\_b\_y), 및 MV\_C\_i는 인접한 C 영역의 블록의 움직임 벡터인 (mv\_c\_x, mv\_c\_y)를 나타내며, 이와 같이 i는 수평 또는 수직 방향을 의미한다. 또한, 임의의 인접 블록이 영상 영역 내에 존재하지 않는 경우는 절대값 함수의 값을 '0'으로 간주한다.

<41> 그런데, 상술한 방법으로 추정된 max\_MV\_E\_i 는 현재 부호화하고자 하는 매크로 블록 E의 움직임 벡터의 움직임이 인접한 블록과 매우 상이한 경우에는 추정한 움직임 벡터값의 오류가 심각하게 된다. 이러한 문제를 방지하기 위해 움직임에 대한 통계적인 특성을 나타내는 최소 탐색 영역값을 구한다(S606). 이 값은 수학식 5와 같이 표현할 수 있다.

<42> 【수학식 5】

$$k_i = \begin{cases} (\text{input\_search\_range} + 2) \div 4 & \text{if}(\alpha_i = 0) \\ 3 \times (\text{input\_search\_range} + 8) \div 16 & \text{if}(0 < \alpha_i \leq 2) \\ (\text{input\_search\_range} + 4) \div 8 & \text{otherwise} \end{cases}$$

$$\alpha_i = \text{abs}(\text{MV\_A\_i}) + \text{abs}(\text{MV\_B\_i}) + \text{abs}(\text{MV\_C\_i})$$

<43> 현재 부호화하고자 하는 매크로 블록 E의 움직임 벡터의 탐색 영역값이

input\_search\_range 값보다 큰 값으로 결정된 경우, 그 값은 오류값이므로 이러한 오류를 방지하기 위해 input\_search\_range를 설정한다. 이러한 input\_search\_range 값은 인접한 블록의 움직임 벡터의 크기에 따라 사용자가 다르게 정의하는 값이다. 그리고 k\_i는 현재 부호화하고자 하는 매크로 블록 E의 움직임 벡터가 가지는 최소 탐색 영역값을 나타내는 것으로, 사용자가 정의하는 input\_search\_range 값과 통계적인 특성을 이용하여 실험을 통해 도출된 오류 방지용 변수이다.

- <44> 수학식 5에서 나타낸 것과 같이,  $k_i$  값은  $a_i$  에 따라 그 값이 달라진다.  $a_i$  가 크다는 것은 인접한 블록의 움직임 벡터의 크기의 합이 크다는 것을 의미한다. 따라서  $a_i$  값이 커지면 사용자는 `input_search_range` 값을 크게 설정해야 하고 그에 따라  $k_i$  값도 커지게 된다. 수학식 5에서 보여지는 상수값들은 인접한 블록의 움직임에 대한 통계적인 특성에 따른 실험값으로써 최적화한 결과이다.
- <45> 수학식 4에서 구한 현재 부호화하고자 하는 매크로 블록 E의 적응 움직임 벡터의 최대 탐색 영역값과 수학식 5에서 구한 최소 탐색 영역값을 비교하여(S608) 현재 부호화하고자 하는 매크로 블록 E의 적응 움직임 벡터의 탐색 영역을 결정한다(S610 및 S612). 단계 S608내지 S612의 동작은 수학식 6과 같이 표현할 수 있다.
- <46> 【수학식 6】  $local\_search\_range\_i = \max(k_i, 2 * max\_MV\_E\_i)$
- <47> 여기서 `local_search_range_i` 는 현재 부호화하고자 하는 매크로 블록 E의 적응 움직임 벡터의 탐색 영역을 나타낸다. `max_MV_E_i` 값에 2를 곱하는 이유는 현재 부호화하고자 하는 매크로 블록 E의 적응 움직임 벡터의 크기가 아무리 크다고 해도 인접 블록의 움직임 벡터의 2배 크기 이상은 되지 않는다는 통계치를 실험을 통해 얻었기 때문이다.
- <48> 수학식 6은  $2 * max\_MV\_E\_i$  값이 수학식 5에서 구한 최소 탐색 영역값보다도 작은 경우에, 현재 부호화하고자 하는 매크로 블록 E의 움직임 벡터의 탐색 영역을 수학식 5에서 구한 최소 탐색 영역값으로 함으로써 움직임 벡터의 오류를 방지하기 위한 것이다. 이는, 현재 부호화하고자 하는 매크로 블록 E의 움직임 벡터의 탐색 영역을 매우 작은 값의 `max_MV_E_i` 값 대신에  $a_i$  값을 고려한 최소 탐색 영역값으로 결정함으로써 움직임 벡터의 오류를 방지하기 위함이다.



<49> 그런데 수학적식 6에서 결정한 현재 부호화하고자 하는 매크로 블록 E의 움직임 벡터의 탐색 영역이 사용자가 정의한 영역보다 크게 설정될 수 있다. 이는 인접한 블록의 움직임 벡터의 크기가 인코딩 시작시 사용자가 디폴트 값으로 정해놓은 값보다 큰 경우 발생한다. 이처럼 수학적식 6에서 결정한 현재 부호화하고자 하는 매크로 블록 E의 움직임 벡터의 탐색 영역값이 사용자가 디폴트 값으로 정해놓은 값보다도 크게 되면, 결정된 값은 아무 의미 없는 값으로서 오히려 이 값을 사용하게 되면 오류가 발생하게 된다. 따라서 이러한 오류를 방지하기 위해 현재 부호화하고자 하는 매크로 블록 E가 가질 수 있는 움직임 벡터의 최대 탐색 영역을 정의하기 위한 제약 조건을 부여한다(S614 내지 S618). 단계 S614 내지 S618 과정은 수학적식 7과 같이 표현할 수 있다.

<50> 【수학적식 7】 
$$\text{new\_search\_range\_i} = \min(\text{input\_search\_range}, \text{local\_search\_range\_i})$$

<51> 전술한 수학적식 3 및 수학적식 7이 현재 부호화하고자 하는 매크로 블록의 최종 적응 움직임 벡터의 탐색 영역을 정하는 모델의 결론이다. 즉, 사용 가능한 인접 블록의 수가 1개 이하인 경우에는 탐색의 신뢰도를 고려하여 최종 적응 움직임 벡터의 탐색 영역을 인코딩 시작시 사용자가 디폴트 값으로 정해놓은 값으로 하고, 사용 가능한 인접 블록의 수가 2개 이상인 경우에는 인접 블록의 움직임 벡터로부터 추정하되 오류 방지를 위해 사용자가 디폴트 값으로 정해놓은 값을 고려하여 최종 적응 움직임 벡터의 탐색 영역을 구한다.

<52> 이상의 설명은 본 발명을 예시적으로 설명한 것에 불과한 것으로, 본 발명이 속하는 기술분야에서 통상의 지식을 가지는 자라면 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 다양한 변형이 가능할 것이다. 따라서, 본 명세서에 개시된 실시예들은 본 발명을 한정하기 위한 것이 아니라 설명하기 위한 것이고, 이러한 실시예에 의하여 본 발명의 사상과 범위가

한정되는 것은 아니다. 본 발명의 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술은 본 발명의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

#### 【발명의 효과】

- <53> 이상에서 설명한 바와 같이 본 발명에 의하면, FMBA 방법에서 필요로 하는 계산량을 줄일 수 있어 효율적인 압축 부호화를 가능하게 하는 효과가 있다. 또한, 인접한 블록의 움직임 벡터들의 움직임으로부터 현재 블록의 움직임 벡터의 탐색 영역을 추정함으로써 종전의 방법보다 더 신뢰성 있는 결과를 제공한다는 효과가 있다. 게다가 본 발명에서는 모드 정보와 같은 데이터를 별도로 요구하지 않고, 단지 인접한 블록의 움직임 벡터들의 크기값만을 요구함으로써 데이터량에 부담이 없다는 장점이 있다. 본 발명은 JVT 동영상 압축 기법이 사용되는 부호화기에서 고속으로 부호화가 가능하도록 하며, 특히 저비트율의 데이터를 압축 부호화, 또는 고속 처리를 요망하는 압축 부호화의 경우에 보다 효율적으로 적용할 수 있다는 장점이 있다.

## 【특허청구범위】

## 【청구항 1】

화상의 연속성을 나타내는 입력 화상 신호가 공급되며 상기 입력 화상 신호의 각각의 화상을 다수의 매크로 블록으로 분할하여 상기 매크로 블록의 움직임 벡터의 움직임 추정을 통해 영상을 부호화하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법에 있어서,

(a) 현재 매크로 블록에 인접한 인접 블록의 수를 판단하는 단계;

(b) 상기 단계 (a)에서의 판단 결과, 상기 인접 블록의 수가 2개 이상인 경우, 상기 인접 블록의 움직임 벡터의 크기를 구하여 그 중 움직임이 가장 큰 블록의 움직임 벡터를 결정하는 단계;

(c) 상기 현재 매크로 블록의 상기 적응 움직임 벡터가 가질 수 있는 최소 탐색 영역값을 정의하는 단계;

(d) 상기 단계 (b)에서 결정된 상기 움직임이 가장 큰 블록의 움직임 벡터 크기의 두 배와 상기 단계 (c)에서 구한 상기 적응 움직임 벡터의 최소 탐색 영역값을 비교하여 더 큰 값을 상기 적응 움직임 벡터의 탐색 영역값으로 결정하는 단계; 및

(e) 상기 단계 (d)에서 구한 상기 적응 움직임 벡터의 탐색 영역값과 사용자가 정의한 적응 움직임 벡터의 탐색 영역값을 비교하여 더 작은 값을 최종 적응 움직임 벡터의 탐색 영역값으로 결정하는 단계

를 포함하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법.

**【청구항 2】**

제 1 항에 있어서, 상기 단계 (b)에서,

상기 인접 블록의 움직임 벡터의 크기는 상기 인접 블록의 움직임 벡터의 수평 및 수직 성분을 검출함으로써 구해지는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법.

**【청구항 3】**

제 1 항에 있어서, 상기 단계 (c)에서,

상기 적응 움직임 벡터의 최소 탐색 영역값은 상기 인접 블록의 움직임 벡터의 크기를 더한 값에 따라 달라지는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법.

**【청구항 4】**

제 3 항에 있어서,

상기 인접 블록의 움직임 벡터의 크기를 더한 값이 0인 경우, 상기 적응 움직임 벡터의 최소 탐색 영역값은, 상기 사용자가 정의한 적응 움직임 벡터의 탐색 영역값에 일정 상수를 더한 값을 4로 나눈 값으로 하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법.

**【청구항 5】**

제 4 항에 있어서,

상기 일정 상수를 2로 하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법.

**【청구항 6】**

제 3 항에 있어서,

상기 인접한 블록의 상기 움직임 벡터의 크기를 더한 값이 0보다 크고 2보다 작거나 같은 경우, 상기 적응 움직임 벡터의 최소 탐색 영역값은, 상기 사용자가 정의한 적응 움직임 벡터의 탐색 영역값에 일정 상수를 더하고 3을 곱하여 16으로 나눈 값으로 하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법.

**【청구항 7】**

제 6 항에 있어서,

상기 일정 상수를 8로 하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법.

**【청구항 8】**

제 3 항에 있어서,

상기 인접한 블록의 상기 움직임 벡터의 크기를 더한 값이 2보다 큰 경우, 상기 적응 움직임 벡터의 최소 탐색 영역값은, 상기 사용자가 정의한 적응 움직임 벡터의 탐색 영역값에 일정 상수를 더한 값을 8로 나눈 값으로 하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법.

**【청구항 9】**

제 8 항에 있어서,

상기 일정 상수를 4로 하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법.

## 【청구항 10】

제 1 항에 있어서,

상기 사용자가 정의한 적응 움직임 벡터의 탐색 영역값은 인코딩 시작시 사용자가 오류 방지를 위해 정한 디폴트 값을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법.

## 【청구항 11】

제 1 항에 있어서,

상기 단계 (a)에서의 판단 결과, 상기 인접한 블록의 수가 1개 이하인 경우, 상기 사용자가 정의한 적응 움직임 벡터의 탐색 영역값을 상기 최종 적응 움직임 벡터의 탐색 영역값으로 결정하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 방법.

## 【청구항 12】

화상의 연속성을 나타내는 입력 화상 신호가 공급되며 상기 입력 화상 신호의 각각의 화상을 다수의 매크로 블록으로 분할하여 상기 매크로 블록의 움직임 벡터의 움직임 추정을 통해 영상을 부호화하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 영상 압축 장치에 있어서,

현재 매크로 블록에 인접한 블록의 수를 판단하는 판단부;

상기 매크로 블록의 움직임 벡터의 크기 또는 상기 현재 매크로 블록의 상기 적응 움직임 벡터가 가질 수 있는 최소 탐색 영역값을 계산하는 계산부;

상기 계산부에서 계산된 인접한 블록들의 움직임 벡터 중 그 크기가 가장 큰 움직임 벡터의 크기의 두 배와 상기 계산부에서 계산한 상기 적응 움직임 벡터의 최소 탐색 영역값을 비교하는 제 1 비교부;

상기 제 1 비교부에서의 비교 결과, 더 큰 값을 상기 적응 움직임 벡터의 탐색 영역값으로 결정하는 결정부;

상기 계산부에서 계산한 상기 적응 움직임 벡터의 탐색 영역값과 사용자가 정의한 적응 움직임 벡터의 탐색 영역값을 비교하는 제 2 비교부; 및

상기 제 2 비교부에서의 비교 결과, 더 작은 값을 최종 적응 움직임 벡터의 탐색 영역값으로 결정하는 최종 결정부

를 포함하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 영상 압축 장치.

#### 【청구항 13】

제 12 항에 있어서,

상기 움직임 벡터의 크기는 상기 움직임 벡터의 수평 및 수직 성분을 검출함으로써 구해지는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 영상 압축 장치.

#### 【청구항 14】

제 12 항에 있어서,

상기 계산부는 상기 인접 블록들의 상기 움직임 벡터의 크기를 더한 값에 따라 상기 적응 움직임 벡터의 최소 탐색 영역값을 다르게 계산하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 영상 압축 장치.

【청구항 15】

제 14 항에 있어서,

상기 인접 블록의 상기 움직임 벡터의 크기를 더한 값이 0인 경우, 상기 적응 움직임 벡터의 최소 탐색 영역값은, 상기 사용자가 정의한 적응 움직임 벡터의 탐색 영역값에 일정 상수를 더한 값을 4로 나눈 값으로 하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 영상 압축 장치.

【청구항 16】

제 15 항에 있어서,

상기 일정 상수를 2로 하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 영상 압축 장치.

【청구항 17】

제 14 항에 있어서,

상기 인접한 블록의 상기 움직임 벡터의 크기를 더한 값이 0보다 크고 2보다 작거나 같은 경우, 상기 적응 움직임 벡터의 최소 탐색 영역값은, 상기 사용자가 정의한 적응 움직임 벡터의 탐색 영역값에 일정 상수를 더하고 3을 곱하여 16으로 나눈 값으로 하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 영상 압축 장치.



**【청구항 18】**

제 17 항에 있어서,

상기 일정 상수를 8로 하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 영상 압축 장치.

**【청구항 19】**

제 14 항에 있어서,

상기 인접한 블록의 상기 움직임 벡터의 크기를 더한 값이 2보다 큰 경우, 상기 적응 움직임 벡터의 최소 탐색 영역값은, 상기 사용자가 정의한 적응 움직임 벡터의 탐색 영역값에 일정 상수를 더한 값을 8로 나눈 값으로 하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 영상 압축 장치.

**【청구항 20】**

제 19 항에 있어서,

상기 일정 상수를 4로 하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 영상 압축 장치.

**【청구항 21】**

제 12 항에 있어서,

상기 사용자가 정의한 적응 움직임 벡터의 탐색 영역값은 인코딩 시작시 사용자가 오류 방지를 위해 정한 디폴트 값을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 영상 압축 장치.

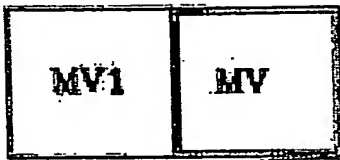
## 【청구항 22】

제 12 항에 있어서,

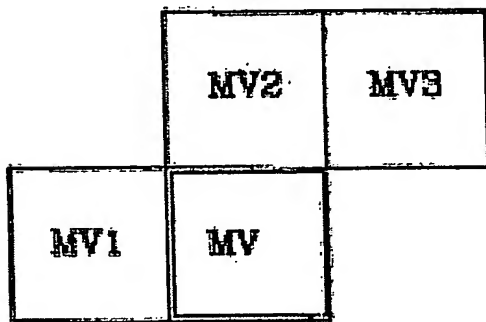
상기 최종 결정부는 상기 판단부에서의 판단 결과 상기 인접한 블록의 수가 1개 이하인 경우, 상기 사용자가 정의한 적응 움직임 벡터의 탐색 영역값을 상기 최종 적응 움직임 벡터의 탐색 영역값으로 결정하는 것을 특징으로 하는 비디오 부호기에서 적응 움직임 벡터의 탐색 영역을 결정하는 영상 압축 장치.

【도면】

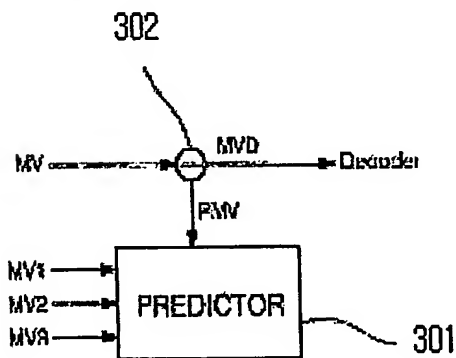
【도 1】



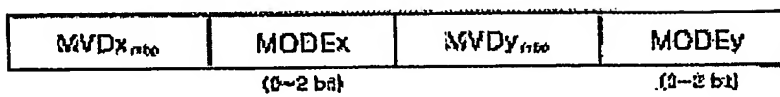
【도 2】



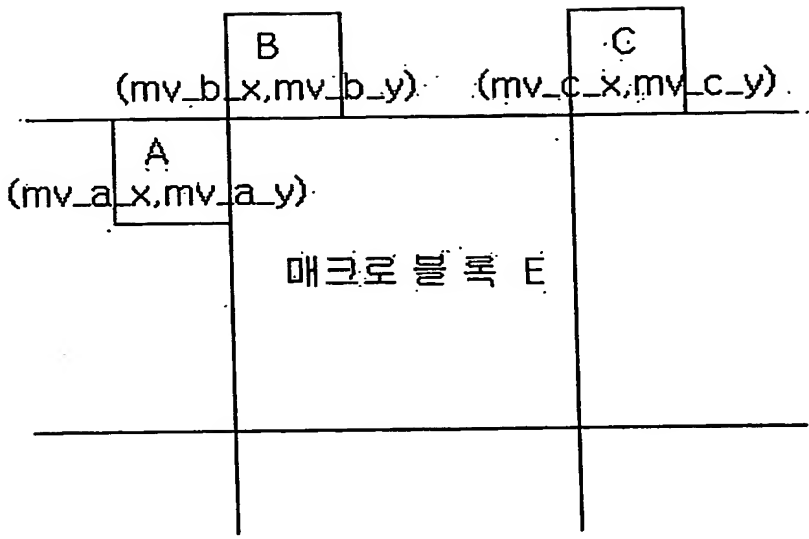
【도 3】



【도 4】



【도 5】



【도 6】

